





























































































**Tabel 4.1 Contoh data latih manualisasi**

PENYAKIT	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17
P1	8	0	6	0	8	0	0	0	0	0	0	0	0	0	0	0	0
P1	8	7	6	8	0	5	0	0	0	0	0	0	0	0	0	0	0
P1	8	7	0	8	8	0	0	0	0	0	0	0	0	0	0	0	0
P2	0	0	0	8	0	0	0	6	8	0	0	0	0	0	0	0	0
P2	7	0	0	8	0	0	7	6	8	0	0	0	0	0	0	0	0
P2	6	0	0	8	0	0	0	6	8	5	0	0	0	0	0	0	0
P3	0	0	0	0	0	0	0	0	0	0	6	7	0	0	0	0	0
P3	0	0	0	0	0	0	0	0	0	0	0	6	7	0	0	0	0
P3	0	0	0	0	0	0	0	0	0	0	8	0	7	0	0	0	0
P4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7
P4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	8
P4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8
P5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8
P5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	8
P5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	8

Pad  
uji yang d

uji saja, data

PENYAKIT	G1
??	

G15	G16	G17
0	0	0

#### 4.3.1 Hit

Tahap kedua adalah penghitungan jarak setiap data latih terhadap data uji menggunakan persamaan 2-3.

Contoh :

Data latih :

1	8	0	6	0	8	0	0	0	0	0	0	0	0	0	0	0	0
2	8	7	6	8	0	5	0	0	0	0	0	0	0	0	0	0	0
3	8	7	0	8	8	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	8	0	0	0	6	8	0	0	0	0	0	0	0	0
5	7	0	0	8	0	0	7	6	8	0	0	0	0	0	0	0	0
6	6	0	0	8	0	0	0	6	8	5	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	6	7	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	6	7	0	0	0	0

[illegible]

Data Uji :

?? 7 7 0 6 0 0 0 8 0 0 0 0 0 0 0 0 0

Hitung :

$$D_1 = \sqrt{(8-7)^2 + (0-7)^2 + (6-0)^2 + (0-6)^2 + (8-0)^2 + (0-0)^2 + (0-0)^2 + (0-8)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2}$$

$$D_1 = 16$$

$$D_2 = \sqrt{(8-7)^2 + (7-7)^2 + (6-0)^2 + (8-6)^2 + (8-0)^2 + (0-0)^2 + (5-0)^2 + (0-8)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2}$$

$$D_2 = 11$$

[illegible]

$$D_{15} = \sqrt{(0-7)^2 + (0-7)^2 + (0-0)^2 + (0-6)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-8)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (8-0)^2 + (0-0)^2 + (8-0)^2}$$

$$D_{15} = 18$$

Hasil Perhitungan euclidean distance ditunjukkan oleh table 4.3.

**Tabel 4.3 Hasil perhitungan euclidean distance**

[illegible]

P1	8	7	0	8	8	0	0	0	0	0	0	0	0	0	0	0	0	12
P2	0	0	0	8	0	0	0	6	8	0	0	0	0	0	0	0	0	13
P2	7	0	0	8	0	0	7	6	8	0	0	0	0	0	0	0	0	13
P2	6	0	0	8	0	0	0	6	8	5	0	0	0	0	0	0	0	12
P3	0	0	0	0	0	0	0	0	0	0	6	7	0	0	0	0	0	17
P3	0	0	0	0	0	0	0	0	0	0	0	6	7	0	0	0	0	17
P3	0	0	0	0	0	0	0	0	0	0	8	0	7	0	0	0	0	18
P4	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	7	17
P4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	8	18
P4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	16
P5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	16
P5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	8	18
P5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	8	18

#### 4.3.2 Sorting Berdasarkan Euclidian Distance

Tahap Kedua adalah dilakukan sorting data latih mulai dari nilai euclidian distance yang terkecil hingga yang terbesar. Data sebelum disorting ditunjukkan oleh tabel 4.4 dan Hasil Sorting ditunjukkan oleh tabel 4.5.

Contoh :

Data sebelum dilakukan sorting :

**Tabel 4.4 Data sebelum disorting**

PENYAKIT	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	Jarak
P1	8	0	6	0	8	0	0	0	0	0	0	0	0	0	0	0	0	16
P1	8	7	6	8	0	5	0	0	0	0	0	0	0	0	0	0	0	11
P1	8	7	0	8	8	0	0	0	0	0	0	0	0	0	0	0	0	12
P2	0	0	0	8	0	0	0	6	8	0	0	0	0	0	0	0	0	13
P2	7	0	0	8	0	0	7	6	8	0	0	0	0	0	0	0	0	13
P2	6	0	0	8	0	0	0	6	8	5	0	0	0	0	0	0	0	12
P3	0	0	0	0	0	0	0	0	0	0	6	7	0	0	0	0	0	17
P3	0	0	0	0	0	0	0	0	0	0	0	6	7	0	0	0	0	17
P3	0	0	0	0	0	0	0	0	0	0	8	0	7	0	0	0	0	18
P4	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	7	17



P4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	8	18
P4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	16
P5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	16
P5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	8	18
P5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	8	18

**Tabel 4.5 Data yang telah disorting**

PENYAKIT	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	Jarak
P1	8	7	6	8	0	5	0	0	0	0	0	0	0	0	0	0	0	11
P1	8	7	0	8	8	0	0	0	0	0	0	0	0	0	0	0	0	12
P2	6	0	0	8	0	0	0	6	8	5	0	0	0	0	0	0	0	12
P2	0	0	0	8	0	0	0	6	8	0	0	0	0	0	0	0	0	13
P2	7	0	0	8	0	0	7	6	8	0	0	0	0	0	0	0	0	13
P1	8	0	6	0	8	0	0	0	0	0	0	0	0	0	0	0	0	16
P4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	16
P5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	16
P3	0	0	0	0	0	0	0	0	0	0	6	7	0	0	0	0	0	17
P3	0	0	0	0	0	0	0	0	0	0	0	6	7	0	0	0	0	17
P4	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	7	17
P3	0	0	0	0	0	0	0	0	0	0	8	0	7	0	0	0	0	18
P4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	8	18
P5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	8	18
P5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	8	18

#### 4.3.3 Pemilihan data sebanyak nilai K

Setelah data disorting maka akan masuk ke tahapan selanjutnya yaitu pemilihan dan pengambilan data sejumlah k, pada contoh perhitungan manual kali ini nilai k adalah 5. Tabel data sebelum diambil sebanyak nilai K ditunjukkan oleh tabel 4.6 dan tabel data setelah diambil sebanyak nilai K ditunjukkan oleh tabel 4.7.

**Tabel 4.6 Data sebelum diambil sejumlah K**

PENYAKIT	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	Jarak
P1	8	7	6	8	0	5	0	0	0	0	0	0	0	0	0	0	0	11
P1	8	7	0	8	8	0	0	0	0	0	0	0	0	0	0	0	0	12

P2	6	0	0	8	0	0	0	6	8	5	0	0	0	0	0	0	12
P2	0	0	0	8	0	0	0	6	8	0	0	0	0	0	0	0	13
P2	7	0	0	8	0	0	7	6	8	0	0	0	0	0	0	0	13
P1	8	0	6	0	8	0	0	0	0	0	0	0	0	0	0	0	16
P4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16
P5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16
P3	0	0	0	0	0	0	0	0	0	0	6	7	0	0	0	0	17
P3	0	0	0	0	0	0	0	0	0	0	0	6	7	0	0	0	17
P4	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	7	17
P3	0	0	0	0	0	0	0	0	0	8	0	7	0	0	0	0	18
P4	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	8	18
P5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	8	18
P5	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	8	18

Tabel 4.7 Data yang diambil sejumlah K

PENYAKIT	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	Jarak
P1	8	7	6	8	0	5	0	0	0	0	0	0	0	0	0	0	0	11
P1	8	7	0	8	8	0	0	0	0	0	0	0	0	0	0	0	0	12
P2	6	0	0	8	0	0	0	6	8	5	0	0	0	0	0	0	0	12
P2	0	0	0	8	0	0	0	6	8	0	0	0	0	0	0	0	0	13
P2	7	0	0	8	0	0	7	6	8	0	0	0	0	0	0	0	0	13

#### 4.3.4 Fuzzyfikasi

Setelah proses penentuan nilai k maka akan dilakukan proses fuzzy dimana proses tersebut bertujuan untuk mencari nilai keanggotaan data uji pada masing - masing kelas penyakit dengan menggunakan persamaan 2-4. Untuk itu perlu diketahui nilai n (jumlah data latih) = 15, kemudian kelas P1 sebanyak 3, kelas P2 sebanyak 3, kelas P3 sebanyak 3, kelas P4 sebanyak 3, kelas P5 sebanyak 3, dimana komponen tersebut diperlukan untuk mendapatkan nilai *membership* pada tiap kelas.

$$P1y = 0,51 + ((3/15)*0,49)$$

$$= 0,6$$

$$P1n = (3/15)*0,49$$

$$= 0,1$$

$$P2y = 0,51 + ((3/15)*0,49)$$

$$= 0,6$$

$$P2n = (3/15)*0,49$$

$$= 0,1$$

$$P3y = 0,51 + ((3/15)*0,49)$$

$$= 0,6$$

$$P3n = (3/15)*0,49$$

$$= 0,1$$

$$P4y = 0,51 + ((3/15)*0,49)$$

$$= 0,6$$

$$P4n = (3/15)*0,49$$

$$= 0,1$$

$$P5y = 0,51 + ((3/15)*0,49)$$

$$= 0,6$$

$$P5n = (3/15)*0,49$$

$$= 0,1$$

#### 4.3.5 Defuzzyfikasi

Setelah didapatkan nilai *membership* untuk masing – masing kelas, dilakukan perhitungan nilai keanggotaan data uji pada tiap kelas menggunakan persamaan 2-5, dimana pada perhitungan di sini digunakan nilai  $m = 2$ .

$$P1 = \frac{\left(0,6\left(\frac{-2}{11^2-1}\right)\right) + \left(0,6\left(\frac{-2}{12^2-1}\right)\right) + \left(0,1 + \left(\frac{-2}{12^2-1}\right)\right) + \left(0,1\left(\frac{-2}{13^2-1}\right)\right) + \left(0,1\left(\frac{-2}{13^2-1}\right)\right)}{\left(\frac{-2}{11^2-1}\right) + \left(\frac{-2}{12^2-1}\right) + \left(\frac{-2}{12^2-1}\right) + \left(\frac{-2}{13^2-1}\right) + \left(\frac{-2}{13^2-1}\right)}$$

$$P1 = 0,326$$

$$P2 = \frac{\left(0,1\left(\frac{-2}{11^2-1}\right)\right) + \left(0,1\left(\frac{-2}{12^2-1}\right)\right) + \left(0,6 + \left(\frac{-2}{12^2-1}\right)\right) + \left(0,6\left(\frac{-2}{13^2-1}\right)\right) + \left(0,6\left(\frac{-2}{13^2-1}\right)\right)}{\left(\frac{-2}{11^2-1}\right) + \left(\frac{-2}{12^2-1}\right) + \left(\frac{-2}{12^2-1}\right) + \left(\frac{-2}{13^2-1}\right) + \left(\frac{-2}{13^2-1}\right)}$$

$$P2 = 0,380$$

$$P3 = \frac{\left(0,1\left(\frac{-2}{11^2-1}\right)\right) + \left(0,1\left(\frac{-2}{12^2-1}\right)\right) + \left(0,1 + \left(\frac{-2}{12^2-1}\right)\right) + \left(0,1\left(\frac{-2}{13^2-1}\right)\right) + \left(0,1\left(\frac{-2}{13^2-1}\right)\right)}{\left(\frac{-2}{11^2-1}\right) + \left(\frac{-2}{12^2-1}\right) + \left(\frac{-2}{12^2-1}\right) + \left(\frac{-2}{13^2-1}\right) + \left(\frac{-2}{13^2-1}\right)}$$

$$P3 = 0,098$$

$$P4 = \frac{\left(0,1\left(\frac{-2}{11^2-1}\right)\right) + \left(0,1\left(\frac{-2}{12^2-1}\right)\right) + \left(0,1 + \left(\frac{-2}{12^2-1}\right)\right) + \left(0,1\left(\frac{-2}{13^2-1}\right)\right) + \left(0,1\left(\frac{-2}{13^2-1}\right)\right)}{\left(\frac{-2}{11^2-1}\right) + \left(\frac{-2}{12^2-1}\right) + \left(\frac{-2}{12^2-1}\right) + \left(\frac{-2}{13^2-1}\right) + \left(\frac{-2}{13^2-1}\right)}$$

$$P4 = 0,098$$

$$P5 = \frac{\left(0,1\left(\frac{-2}{11^2-1}\right)\right) + \left(0,1\left(\frac{-2}{12^2-1}\right)\right) + \left(0,1 + \left(\frac{-2}{12^2-1}\right)\right) + \left(0,1\left(\frac{-2}{13^2-1}\right)\right) + \left(0,1\left(\frac{-2}{13^2-1}\right)\right)}{\left(\frac{-2}{11^2-1}\right) + \left(\frac{-2}{12^2-1}\right) + \left(\frac{-2}{12^2-1}\right) + \left(\frac{-2}{13^2-1}\right) + \left(\frac{-2}{13^2-1}\right)}$$

$$P5 = 0,098$$

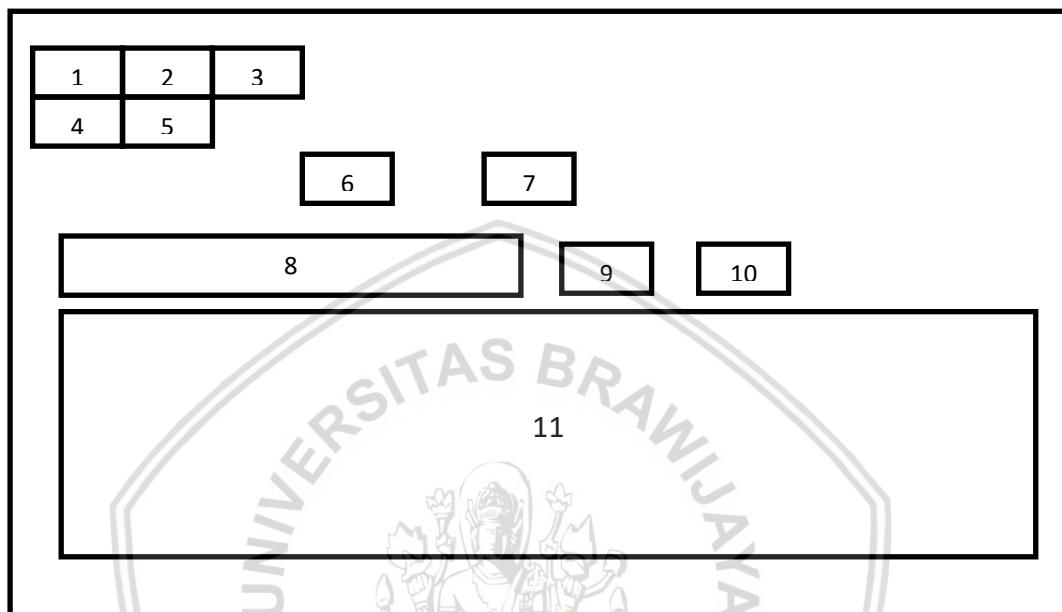
Dari perhitungan tersebut didapatkan nilai keanggotaan dari tiap kelas. Dimana akhirnya didapatkan untuk nilai keanggotaan data uji (x) untuk kelas P1 bernilai = 0,326. Sedangkan untuk untuk nilai keanggotaan data uji (x) pada kelas P2 didapatkan nilai 0,380. Sedangkan untuk untuk nilai keanggotaan data uji (x) pada kelas P3 didapatkan nilai 0,098. Sedangkan untuk untuk nilai keanggotaan data uji (x) pada kelas P4 didapatkan nilai 0,098. Sedangkan untuk untuk nilai keanggotaan data uji (x) pada kelas P5 didapatkan nilai 0,098. Dari perhitungan tersebut dapat disimpulkan bahwa nilai peluang terbesar dimiliki oleh P2, sehingga dapat disimpulkan bahwa data uji yang dimasukkan termasuk dalam P2.

#### 4.4 Perancangan Antarmuka

Perancangan antarmuka dilakukan untuk memudahkan pembuatan antarmuka di mana antarmuka sendiri merupakan penghubung antara pengguna dengan sistem. Sistem diagnosis penyakit tanaman cabai yang akan dibangun merupakan sistem menggunakan bahasa pemrograman java. Sistem akan memberikan keluaran berupa hasil perhitungan-perhitungan proses FK-NN dan hasil diagnosis penyakit tanaman cabai berdasarkan masukan yang diberikan yang berasal dari pengguna. Masukan-masukan tersebut berupa gejala penyakit. Masukan dari pengguna kemudian dibaca oleh sistem dan dilakukan perhitungan menggunakan FK-NN.

#### 4.4.1 Perancangan Antarmuka Data Latih

Perancangan antarmuka data latih ini adalah halaman awal yang muncul ketika perangkat lunak dijalankan, pada halaman ini pengguna diminta memasukkan data latih yang tersimpan di file .txt. Perancangan antarmuka data latih ditunjukkan oleh gambar 4.5.



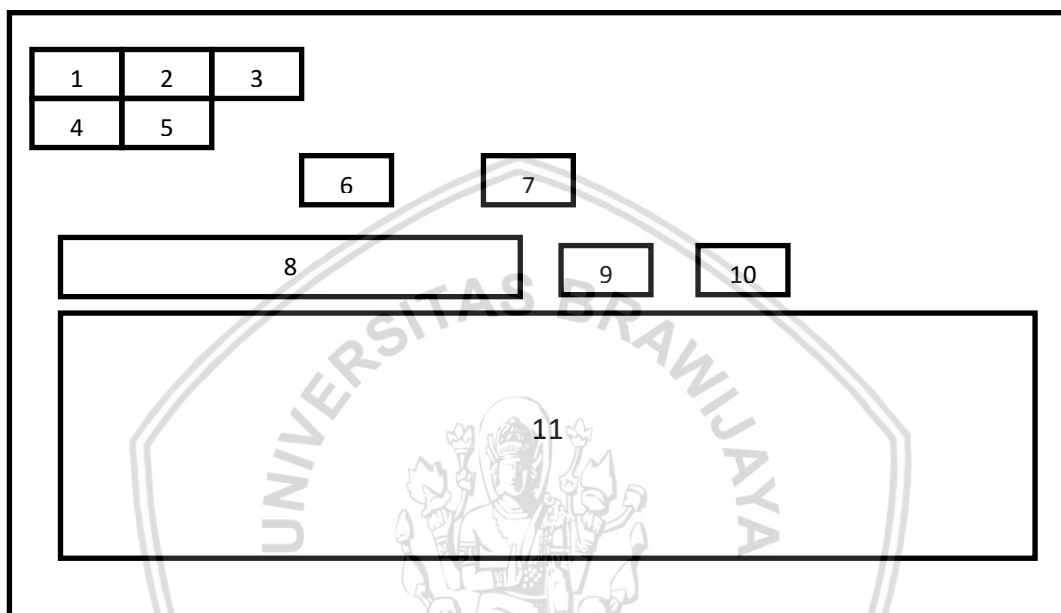
**Gambar 4.5 Perancangan Antarmuka Data Latih**

Ket :

1. Data
2. Proses FKNN
3. Hasil Klasifikasi dan Akurasi
4. Data Latih
5. Data Uji
6. Jumlah Data Latih
7. Jumlah Gejala
8. Lokasi Data Latih
9. Browse
10. Ambil Data
11. List Data Latih

#### 4.4.2 Perancangan Antarmuka Data Uji

Perancangan antarmuka data uji ini adalah halaman yang digunakan untuk memanggil data uji. Pada halaman ini pengguna diminta memasukkan data uji yang tersimpan di file .txt. Perancangan antarmuka data uji ditunjukkan oleh gambar 4.6



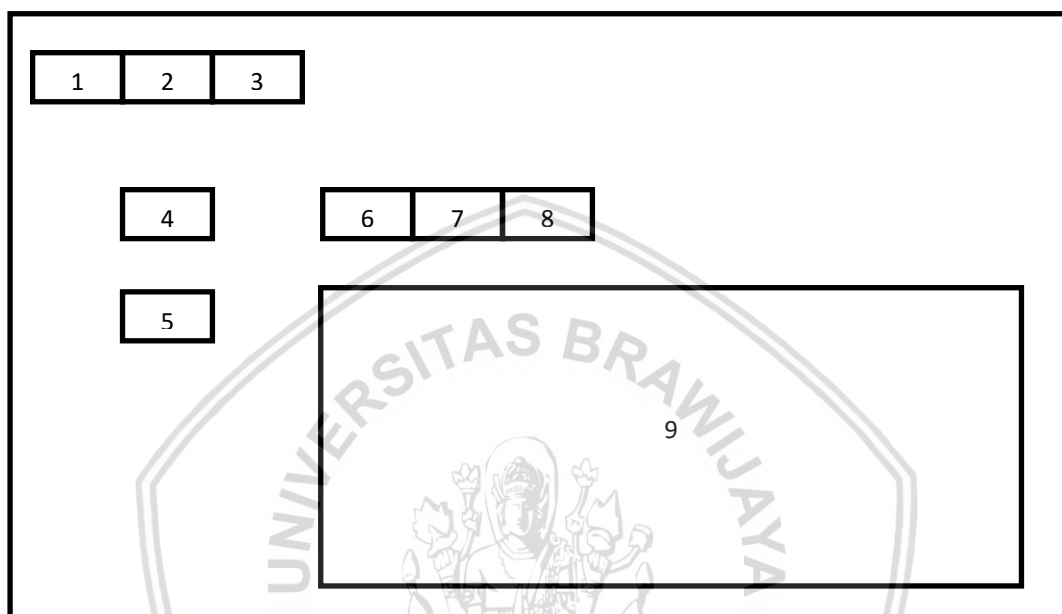
**Gambar 4.6 Perancangan Antarmuka Data Uji**

Ket :

1. Data
2. Proses FKNN
3. Hasil Klasifikasi dan Akurasi
4. Data Latih
5. Data Uji
6. Jumlah Data Latih
7. Jumlah Gejala
8. Lokasi Data Uji
9. Browse
10. Ambil Data
11. List Data Uji

#### 4.4.3 Perancangan Antarmuka Proses FKNN

Perancangan antarmuka FKNN ini adalah halaman dimana akan ditampilkan hasil proses perhitungan FKNN setelah pengguna memasukkan data latih dan data uji disertai dengan nilai K yang dimasukkan pada halaman ini. Halaman antarmuka proses FKNN ditunjukkan oleh Gambar 4.7.



**Gambar 4.7 Perancangan Antarmuka Proses FKNN**

Ket :

1. Data
2. Proses FKNN
3. Hasil Klasifikasi dan Akurasi
4. Nilai K
5. Proses
6. Euclidian
7. K-Data
8. Proses Fuzzy
9. Hasil Perhitungan

## BAB 5 IMPLEMENTASI

Berdasarkan analisis kebutuhan dan perancangan yang telah dilakukan pada sebelumnya, bagian ini akan membahas tentang implementasi algoritma FK-NN untuk diagnosis penyakit tanaman cabai. Bahasan-bahasan dalam bab ini terdiri dari spesifikasi sistem, batasan implementasi, implementasi algoritma, dan implementasi antarmuka.

### 5.1 Implementasi sistem

Implementasi sistem memerlukan dua aspek penting agar sistem dapat berjalan dengan baik sesuai fungsinya. Dua aspek tersebut adalah perangkat keras dan perangkat lunak yang digunakan untuk membangun sistem.

#### 5.1.1 Spesifikasi Perangkat Keras

Tabel 5.1 menunjukkan daftar spesifikasi perangkat keras yang digunakan untuk membangun sistem.

**Tabel 5.1 Spesifikasi Perangkat Keras**

Nama Komponen	Spesifikasi
Prosesor	Intel(R) Core(TM) i3-3217U CPU @ 1,80GHz
Memori	4 GB
Kartu grafis	Intel(R) HD Graphics 4000
Hardisk	500 GB

#### 5.1.2 Spesifikasi Perangkat Lunak

Tabel 5.2 menunjukkan daftar spesifikasi perangkat lunak yang digunakan untuk membangun sistem.

**Tabel 5.2 Spesifikasi Perangkat Lunak**

Nama	Spesifikasi
Sistem operasi	Windows 10
Bahasa pemrograman	Java
Tool pemrograman	Netbeans 8.0.1, JDK

### 5.2 Batasan Implementasi

Adapun batasan-batasan dalam implementasi algoritma FK-NN untuk diagnosis penyakit tanaman cabai yaitu:

1. Sistem yang dibangun menggunakan bahasa pemrograman java.
2. Data yang digunakan dalam sistem disimpan dalam file txt.
3. Data yang digunakan berupa data-data gejala penyakit tanaman cabai,.



4. Keluaran dari sistem adalah salah satu dari 4 penyakit tanaman cabai.
5. Metode yang digunakan adalah fuzzy K-NN.

### 5.3 Implementasi Algoritma

Implementasi algoritme FK-NN untuk diagnosis penyakit tanaman cabai terdiri dari perhitungan jarak Euclidean antara data latih dengan data uji, sorting hasil perhitungan jarak euclidean antara data latih dan data uji, pengambilan hasil perhitungan sebanyak k, fuzzifikasi dan defuzzifikasi.

#### 5.3.1 Implementasi Perhitungan Jarak Euclidean Data Latih dan Data Uji

Langkah berikutnya yaitu menghitung jarak Euclidean antara data latih dengan data uji yang ditunjukkan pada Source Code 5.1.

```

1  public void hitung_jarak (double data_uji[], double data_latih[], String
   kelas[]){
2  this.jarak = new double[data_uji.length][data_latih.length];
3  this.kelas = new String[data_uji.length][data_latih.length];
4
5  for (int i=0; i < jarak.length; i++){
6  for (int j=0; j < jarak[0].length; j++){
7  double tot = 0.0;
8  for (int k=0; k < data_latih[0].length; k++){
9  tot += Math.pow(data_uji[i][k]-data_latih[j][k], 2);
10 }
11 this.jarak[i][j] = Math.sqrt(tot);
12 this.kelas[i][j] = kelas[j];
13 }
14 }
15 }
16
17 public double[][] getJarak() {
18 return jarak;
19 }
20
21 public String[][] getKelas() {
22 return kelas;
23 }

```

**Source Code 5.1 Hitung Euclidean Distance**

Tahap pertama pada implementasi algoritma adalah perhitungan euclidean distance antara data latih dengan data uji, dimana dari perhitungan ini akan didapatkan nilai jarak data uji dengan setiap data latih.

### 5.3.2 Implementasi Pengurutan Jarak Euclidian

Langkah berikutnya yaitu pengurutan jarak Euclidean antara data latih dengan data uji yang ditunjukkan pada Source Code 5.2.

```
1  public void urut_jarak (double jarak[][] , String kelas[]){
2      this.jarak_terurut = copy_jarak(jarak);
3      this.kelas_terurut = copy_kelas(kelas);
4      double temp_jrk = 0.0;
5      String temp_kls = " ";
6
7      for (int i=0; i < jarak_terurut.length; i++){
8          for (int j=0; j < jarak_terurut[0].length-1; j++){
9              for (int k=0; k < jarak_terurut[0].length-1-j; k++){
10
11                  if (this.jarak_terurut[i][k] > this.jarak_terurut[i][k+1]){
12                      temp_jrk = this.jarak_terurut[i][k];
13                      this.jarak_terurut[i][k] = this.jarak_terurut[i][k+1];
14                      this.jarak_terurut[i][k+1] = temp_jrk;
15
16                      temp_kls = this.kelas_terurut[i][k];
17                      this.kelas_terurut[i][k] = this.kelas_terurut[i][k+1];
18                      this.kelas_terurut[i][k+1] = temp_kls;
19
20                  }
21              }
22          }
23      }
24  }
25
26  public double[][] getJarak_terurut() {
27      return jarak_terurut;
28  }
29
30  public String[][] getKelas_terurut() {
31      return kelas_terurut;
32  }
```

**Source Code 5.2 Pengurutan Jarak Euclidian**

Setelah sebelumnya dilakukan penghitungan jarak euclidian data latih terhadap data uji, kemudian dilakukan pengurutan hasil perhitungan jarak euclidian dari nilai terkecil hingga terbesar.

### 5.3.3 Implementasi Pengambilan Hasil Penghitungan Jarak Euclidian Sebanyak K

Langkah berikutnya yaitu Pengambilan hasil penghitungan jarak euclidian sebanyak K yang ditunjukkan pada Source Code 5.3.

```

1  public void ambil_kdata (int K, double jarak_terurut[][], String kelas_terurut[][]){
2      this.k_jarak = new double[jarak_terurut.length][K];
3      this.k_kelas = new String[jarak_terurut.length][K];
4
5      for (int i=0; i < k_jarak.length; i++){
6          for (int j=0; j < k_jarak[0].length; j++){
7              this.k_jarak[i][j] = jarak_terurut[i][j];
8              this.k_kelas[i][j] = kelas_terurut[i][j];
9          }
10     }
11 }
12
13 public double[][] getK_jarak() {
14     return k_jarak;
15 }
16
17 public String[][] getK_kelas() {
18     return k_kelas;
19 }
20
21 public void cek_nilai_jarak_sama(String k_kelas[][]){
22     this.sama_semua = new String[k_kelas.length];
23
24     for (int i=0; i < sama_semua.length; i++){
25         for (int j=1; j < k_kelas[0].length; j++){
26             if (k_kelas[i][j-1].equalsIgnoreCase(k_kelas[i][j])){
27                 this.sama_semua[i] = "Sama Semua";
28             } else {
29                 this.sama_semua[i] = "Tidak Sama";
30                 break;
31             }
32         }
33     }
34
35 }
36
37 public String[] getSama_semua() {

```

```

38     return sama_semua;
39 }

```

**Source Code 5.3 Proses Pengambilan Hasil Penghitungan Jarak Euclidian  
Sebanyak K**

Setelah sebelumnya dilakukan pengurutan jarak euclidian data latih terhadap data uji, kemudian diambil jarak terkecil sebanyak K.

### 5.3.4 Implementasi Fuzzifikasi

Langkah berikutnya yaitu proses fuzzifikasi yang ditunjukkan pada Source Code 5.4.

```

1  public void fuzzyfikasi (String [] kelas_data_latih, int jlh_dt_latih){
2      this.jlh_tiap_kls = hitung_jlh_tiap_kls(kelas_data_latih);
3      this.fuzzyfikasi_y = hitung_py(jlh_tiap_kls, jlh_dt_latih);
4      this.fuzzyfikasi_n = hitung_pn(jlh_tiap_kls, jlh_dt_latih);
5  }
6
7  public double[] getJlh_tiap_kls() {
8      return jlh_tiap_kls;
9  }
10
11 public double[] getFuzzyfikasi_y() {
12     return fuzzyfikasi_y;
13 }
14
15 public double[] getFuzzyfikasi_n() {
16     return fuzzyfikasi_n;
17 }
18 private double[] hitung_jlh_tiap_kls(String[] kelas_data_latih) {
19     double jlh_dt_latih[] = new double[4];
20
21     for (int i=0; i < kelas_data_latih.length; i++){
22         //jumlah penyakit
23         if (kelas_data_latih[i].equalsIgnoreCase("BusukBuah")){
24             jlh_dt_latih[0] += 1;
25         } else
26         if (kelas_data_latih[i].equalsIgnoreCase("BercakDaun")){
27             jlh_dt_latih[1] += 1;
28         } else
29         if (kelas_data_latih[i].equalsIgnoreCase("LayuFusarium")){
30             jlh_dt_latih[2] += 1;
31         } else
32         if (kelas_data_latih[i].equalsIgnoreCase("VirusGemini")){

```

```

33         jlh_dt_latih[3] += 1;
34     }
35 }
36
37     return jlh_dt_latih;
38 }
39
40     private double[] hitung_py(double[] jlh_tiap_kls, int jlh_dt_latih) {
41         double tmp_py[] = new double[jlh_tiap_kls.length];
42
43         for (int i=0; i < jlh_tiap_kls.length; i++){
44             tmp_py[i] = 0.51+((jlh_tiap_kls[i]/jlh_dt_latih)*0.49);
45         }
46
47         return tmp_py;
48     }
49
50     private double[] hitung_pn(double[] jlh_tiap_kls, int jlh_dt_latih) {
51         double tmp_pn[] = new double[jlh_tiap_kls.length];
52
53         for (int i=0; i < jlh_tiap_kls.length; i++){
54             tmp_pn[i] = (jlh_tiap_kls[i]/jlh_dt_latih)*0.49;
55         }
56
57         return tmp_pn;
58     }

```

**Source Code 5.4 Implementasi Fuzzifikasi**

Setelah sebelumnya dilakukan pengambilan jarak euclidian data latih terhadap data uji terkecil sebanyak K, kemudian dilakukan proses perhitungan fuzzifikasi setiap kelas.

### 5.3.5 Implementasi Defuzzifikasi

Langkah berikutnya yaitu proses defuzzifikasi yang ditunjukkan pada Source Code 5.5.

```

private double[][] hitung_pembilang(String[][] k_kls, double[][] k_jarak, double[]
1 fuzzyfikasi_y, double[] fuzzyfikasi_n, int m) {
2     double tmp_defuzzyfikasi[][] = new double[k_kls.length][4];
3
4     for (int i=0; i < k_kls.length; i++){
5         for (int j=0; j < k_kls[0].length; j++){
6
7             //defuzzyfikasi penyakit 1
8             if (k_kls[i][j].equalsIgnoreCase("BusukBuah")){

```

```

9      if (k_jarak[i][j] == 0.0){
10         tmp_defuzzyfikasi[i][0] += 0.0;
11     } else {
12         tmp_defuzzyfikasi[i][0] +=
13         fuzzyfikasi_y[0]*(Math.abs(Math.pow(k_jarak[i][j], (-2/(m-1)))));
14     }
15     } else {
16         if (k_jarak[i][j] == 0.0){
17             tmp_defuzzyfikasi[i][0] += 0.0;
18         } else {
19             tmp_defuzzyfikasi[i][0] +=
20             fuzzyfikasi_n[0]*(Math.abs(Math.pow(k_jarak[i][j], (-2/(m-1)))));
21         }
22     }
23     //defuzzyfikasi penyakit 2
24     if (k_kls[i][j].equalsIgnoreCase("BercakDaun")){
25         if (k_jarak[i][j] == 0.0){
26             tmp_defuzzyfikasi[i][1] += 0.0;
27         } else {
28             tmp_defuzzyfikasi[i][1] +=
29             fuzzyfikasi_y[1]*(Math.abs(Math.pow(k_jarak[i][j], (-2/(m-1)))));
30         }
31     } else {
32         if (k_jarak[i][j] == 0.0){
33             tmp_defuzzyfikasi[i][1] += 0.0;
34         } else {
35             tmp_defuzzyfikasi[i][1] +=
36             fuzzyfikasi_n[1]*(Math.abs(Math.pow(k_jarak[i][j], (-2/(m-1)))));
37         }
38     }
39     //defuzzyfikasi penyakit 3
40     if (k_kls[i][j].equalsIgnoreCase("LayuFusarium")){
41         if (k_jarak[i][j] == 0.0){
42             tmp_defuzzyfikasi[i][2] += 0.0;
43         } else {
44             tmp_defuzzyfikasi[i][2] +=
45             fuzzyfikasi_y[2]*(Math.abs(Math.pow(k_jarak[i][j], (-2/(m-1)))));
46         }
47     } else {
48         if (k_jarak[i][j] == 0.0){
49             tmp_defuzzyfikasi[i][2] += 0.0;
50         } else {
51             tmp_defuzzyfikasi[i][2] +=
52             fuzzyfikasi_n[2]*(Math.abs(Math.pow(k_jarak[i][j], (-2/(m-1)))));
53         }
54     }

```

```

        tmp_defuzzyfikasi[i][2] +=
48  fuzzyfikasi_n[2]*(Math.abs(Math.pow(k_jarak[i][j], (-2/(m-1)))));
49      }
50  }
51
52  //defuzzyfikasi penyakit 4
53  if (k_ks[i][j].equalsIgnoreCase("VirusGemini")){
54      if (k_jarak[i][j] == 0.0){
55          tmp_defuzzyfikasi[i][3] += 0.0;
56      } else {
57          tmp_defuzzyfikasi[i][3] +=
58          fuzzyfikasi_y[3]*(Math.abs(Math.pow(k_jarak[i][j], (-2/(m-1)))));
59      } else {
60          if (k_jarak[i][j] == 0.0){
61              tmp_defuzzyfikasi[i][3] += 0.0;
62          } else {
63              tmp_defuzzyfikasi[i][3] +=
64              fuzzyfikasi_n[3]*(Math.abs(Math.pow(k_jarak[i][j], (-2/(m-1)))));
65          }
66      }
67  }
68  }
69
70  return tmp_defuzzyfikasi;
71  }
72
73  private double[] hitung_penyebut(double[][] k_jarak, int m) {
74      double tmp_penyebut[] = new double[k_jarak.length];
75
76      for (int i=0; i < k_jarak.length; i++){
77          for (int j=0; j < k_jarak[0].length; j++){
78              if (k_jarak[i][j] != 0.0){
79                  tmp_penyebut[i] += Math.abs(Math.pow(k_jarak[i][j], (-2/(m-1)))));
80              }
81          }
82      }
83
84      return tmp_penyebut;
85  }
86
87  private double[][] hitung_defuzzyfikasi(double[][] pembilang, double[]
    penyebut) {

```



```

        double tmp_defuzzyfikasi[][] = new
88 double[pembilang.length][pembilang[0].length];
89
90     for (int i = 0; i < tmp_defuzzyfikasi.length; i++){
91         for (int j=0; j < tmp_defuzzyfikasi[0].length; j++){
92             if (pembilang[i][j] == 0.0 || penyebut[i] == 0.0){
93                 tmp_defuzzyfikasi[i][j] = 0.0;
94             } else {
95                 tmp_defuzzyfikasi[i][j] = pembilang[i][j]/penyebut[i];
96             }
97         }
98     }
99     return tmp_defuzzyfikasi;
100 }
101
102 private String[][] isi_kelas_defuzzyfikasi(String k_kls[][]){
103     String tmp_kls_defuzzyfikasi[][] = new String[k_kls.length][4];
104
105     for (int i=0; i < tmp_kls_defuzzyfikasi.length; i++){
106         tmp_kls_defuzzyfikasi[i][0] = "BusukBuah";
107         tmp_kls_defuzzyfikasi[i][1] = "BercakDaun";
108         tmp_kls_defuzzyfikasi[i][2] = "LayuFusarium";
109         tmp_kls_defuzzyfikasi[i][3] = "VirusGemini";
110     }
111
112     return tmp_kls_defuzzyfikasi;
113 }
114
115 private double[][] copy_defuzzyfikasi(double[][] defuzzyfikasi) {
116     double[][] tmp_deff = new
117     double[defuzzyfikasi.length][defuzzyfikasi[0].length];
118
119     for (int i=0; i < tmp_deff.length; i++){
120         for (int j=0; j < tmp_deff[0].length; j++){
121             tmp_deff[i][j] = defuzzyfikasi[i][j];
122         }
123     }
124
125     return tmp_deff;
126 }
127
128 private String[][] copy_kls_def(String[][] defuzzyfikasi_kelas) {
    String[][] def_kls = new
    String[defuzzyfikasi_kelas.length][defuzzyfikasi_kelas[0].length];

```



```

129
130     for (int i=0; i < def_kls.length; i++){
131         for (int j=0; j < def_kls[0].length; j++){
132             def_kls[i][j] = defuzzyfikasi_kelas[i][j];
133         }
134     }
135
136     return def_kls;
137 }

```

#### Source Code 5.5 Implementasi Defuzzifikasi

Setelah didapatkan nilai jarak euclidian sebanyak K dan nilai fuzzifikasi setiap penyakit, kemudian dilakukan penghitungan defuzzifikasi.

#### 5.3.6 Implementasi Penentuan Kelas

Langkah berikutnya yaitu proses penentuan kelas yang ditunjukan pada Source Code 5.6.

```

    public void urut_defuzzyfikasi(double defuzzyfikasi[], String
1  defuzzyfikasi_kelas[]){
2      this.def_terurut = copy_defuzzyfikasi(defuzzyfikasi);
3      this.def_kls_terurut= copy_kls_def(defuzzyfikasi_kelas);
4      double temp_jrk = 0.0;
5      String temp_kls = " ";
6
7      for (int i=0; i < def_terurut.length; i++){
8          for (int j=0; j < def_terurut[0].length-1; j++){
9              for (int k=0; k < def_terurut[0].length-1-j; k++){
10                 if (this.def_terurut[i][k] < this.def_terurut[i][k+1]){
11                     temp_jrk = this.def_terurut[i][k];
12                     this.def_terurut[i][k] = this.def_terurut[i][k+1];
13                     this.def_terurut[i][k+1] = temp_jrk;
14
15                     temp_kls = this.def_kls_terurut[i][k];
16                     this.def_kls_terurut[i][k] = this.def_kls_terurut[i][k+1];
17                     this.def_kls_terurut[i][k+1] = temp_kls;
18                 }
19             }
20         }
21     }
22 }
23
24 public double[][] getDef_terurut() {
25     return def_terurut;
26 }

```

```

27
28     public String[][] getDef_kls_terurut() {
29         return def_kls_terurut;
30     }
31
32     //7. Hasil klasifikasi
33     public void klasifikasi(double[][] def_terurut, String def_kls_terurut[], String[]
sama_semua, String K_data[]){
34         this.nilai_defuzzyfikasi = new double[def_terurut.length];
35         this.kls_klasifikasi = new String[def_kls_terurut.length];
36
37         for (int i=0; i < def_terurut.length; i++){
38
39             if (sama_semua[i].equalsIgnoreCase("Tidak Sama")){
40                 this.nilai_defuzzyfikasi[i] = def_terurut[i][0];
41                 this.kls_klasifikasi[i] = def_kls_terurut[i][0];
42             } else {
43                 int idx_ = 0;
44                 for (int j=0; j < def_kls_terurut[0].length; j++){
45                     if (def_kls_terurut[i][j].equalsIgnoreCase(K_data[i][0])){
46                         idx_ = j;
47                         break;
48                     }
49                 }
50
51                 this.nilai_defuzzyfikasi[i] = def_terurut[i][idx_];
52                 this.kls_klasifikasi[i] = K_data[i][0];
53             }
54         }
55     }
56 }

```

**Source Code 5.6 Implementasi Penentuan Kelas**

Tahap terakhir dari FKNN adalah penentuan kelas, kelas dengan nilai terbesar yang dijadikan sebagai kelas data uji.

## 5.4 Implementasi Antarmuka

Antarmuka bertujuan untuk memudahkan interaksi pengguna dengan sistem, Implementasi antarmuka terdiri dari tampilan antarmuka awal, data uji, proses FKNN, Penentuan Kelas dan Akurasi.

### 5.4.1 Implementasi Antarmuka Awal

Tampilan awal merupakan tampilan pertama saat pengguna menjalankan aplikasi. Pada halaman awal pengguna diminta memasukkan file data latih dan data uji yang tersimpan dalam format .txt. Tampilan antarmuka awal ditunjukkan oleh gambar 5.1 dan gambar 5.2.

**DIAGNOSA PENYAKIT PADA TANAMAN CABAI**  
MENGUNAKAN METODE FUZZY-KNN (F-KNN)

DATA PROSES FKNN HASIL KLASIFIKASI DAN AKURASI

DATA LATIH DATA UJI

**INPUT PARAMETER DATA** Jumlah Data Latih  Jumlah Gejala

PILIH DATA

No.	G 1	G 2	G 3	G 4	G 5	G 6	G 7	G 8	G 9	G 10	G 11	G 12	G 13	G 14	...	G 18	Kelas
1	4.0	6.0	7.0	8.0	9.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Busuk...
2	4.0	6.0	7.0	8.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Busuk...
3	4.0	6.0	7.0	0.0	9.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Busuk...
4	4.0	6.0	7.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Busuk...
5	4.0	6.0	0.0	8.0	9.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Busuk...
6	0.0	0.0	0.0	0.0	0.0	4.0	7.0	9.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Berca...
7	0.0	0.0	0.0	0.0	0.0	4.0	7.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Berca...
8	0.0	0.0	0.0	0.0	0.0	4.0	0.0	9.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Berca...
9	0.0	0.0	0.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Berca...
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	7.0	9.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Berca...
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0	6.0	8.0	9.0	0.0	0.0	...	4.0	LayuF...
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0	6.0	8.0	9.0	0.0	0.0	...	0.0	LayuF...
13	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0	6.0	8.0	0.0	0.0	0.0	...	4.0	LayuF...
14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0	6.0	8.0	0.0	0.0	0.0	...	0.0	LayuF...
15	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0	6.0	0.0	9.0	0.0	0.0	...	4.0	LayuF...
16	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0	5.0	...	0.0	Virus...
17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0	5.0	...	0.0	Virus...
18	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0	5.0	...	0.0	Virus...
19	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0	5.0	...	0.0	Virus...
20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0	5.0	...	0.0	Virus...

Gambar 5.1 Implementasi Antarmuka Pilih Data Latih

**DIAGNOSA PENYAKIT PADA TANAMAN CABAI**  
MENGUNAKAN METODE FUZZY-KNN (F-KNN)

DATA PROSES FKNN HASIL KLASIFIKASI DAN AKURASI

DATA LATIH DATA UJI

**INPUT PARAMETER DATA** Jumlah Data Uji  Jumlah Gejala

PILIH DATA

No.	G 1	G 2	G 3	G 4	G 5	G 6	G 7	G 8	G 9	G 10	G 11	G 12	G 13	G 14	...	G 18	Kelas
1	4.0	6.0	7.0	8.0	9.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Busu...
2	4.0	6.0	7.0	8.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Busu...
3	4.0	6.0	7.0	0.0	9.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Busu...
4	4.0	6.0	7.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Busu...
5	4.0	6.0	0.0	8.0	9.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Busu...
6	4.0	6.0	0.0	8.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Busu...
7	4.0	6.0	0.0	0.0	9.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Busu...
8	4.0	6.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Busu...
9	4.0	0.0	7.0	8.0	9.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Busu...
10	4.0	0.0	7.0	8.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Busu...
11	4.0	0.0	7.0	0.0	9.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Busu...
12	4.0	0.0	7.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Busu...
13	4.0	0.0	0.0	8.0	9.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Busu...
14	4.0	0.0	0.0	8.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Busu...
15	4.0	0.0	0.0	0.0	9.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Busu...
16	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Busu...
17	0.0	6.0	7.0	8.0	9.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Busu...
18	0.0	6.0	7.0	8.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Busu...
19	0.0	6.0	7.0	0.0	9.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Busu...
20	0.0	6.0	7.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Busu...

Gambar 5.2 Implementasi Antarmuka Pilih Data Uji

## 5.4.2 Implementasi Antarmuka Proses FKNN

Pada implementasi antarmuka proses FKNN pengguna diminta memasukkan nilai K kemudian memilih tombol proses, selanjutnya akan ditampilkan hasil perhitungan fuzzy KNN mulai dari perhitungan jarak Euclidian hingga Defuzzifikasi yang ditunjukkan oleh gambar 5.3 hingga gambar 5.7.

**DIAGNOSA PENYAKIT PADA TANAMAN CABAI**  
MENGUNAKAN METODE FUZZY-KNN (F-KNN)

DATA PROSES FKNN HASIL KLASIFIKASI DAN AKURASI

PARAMETER

NILAI K-

JUMLAH DATA UJI

JUMLAH DATA LATIH

PROSES

D-EUCLIDEAN K-DATA PROSES FUZZY

SEBELUM DIURUT SETELAH DIURUT

[ 78 ]	21,354 ==> BusukBuah	19,365 ==> BusukBuah	19,799 ==> BusukBuah	17,635 ==> BusukBuah	20,174 ==> BusukBuah
[ 79 ]	19,365 ==> BusukBuah	17,146 ==> BusukBuah	17,635 ==> BusukBuah	15,166 ==> BusukBuah	18,055 ==> BusukBuah
[ 80 ]	19,799 ==> BusukBuah	17,635 ==> BusukBuah	18,111 ==> BusukBuah	15,716 ==> BusukBuah	18,52 ==> BusukBuah
[ 81 ]	17,635 ==> BusukBuah	15,166 ==> BusukBuah	15,716 ==> BusukBuah	12,884 ==> BusukBuah	16,186 ==> BusukBuah
[ 82 ]	20,174 ==> BusukBuah	18,055 ==> BusukBuah	18,52 ==> BusukBuah	16,186 ==> BusukBuah	18,921 ==> BusukBuah
[ 83 ]	18,055 ==> BusukBuah	15,652 ==> BusukBuah	16,186 ==> BusukBuah	13,454 ==> BusukBuah	16,643 ==> BusukBuah
[ 84 ]	18,52 ==> BusukBuah	16,186 ==> BusukBuah	16,703 ==> BusukBuah	14,071 ==> BusukBuah	17,146 ==> BusukBuah
[ 85 ]	16,186 ==> BusukBuah	13,454 ==> BusukBuah	14,071 ==> BusukBuah	10,817 ==> BusukBuah	14,595 ==> BusukBuah
[ 86 ]	21,564 ==> BusukBuah	19,596 ==> BusukBuah	20,025 ==> BusukBuah	17,889 ==> BusukBuah	20,396 ==> BusukBuah
[ 87 ]	19,596 ==> BusukBuah	17,407 ==> BusukBuah	17,889 ==> BusukBuah	15,46 ==> BusukBuah	18,303 ==> BusukBuah
[ 88 ]	20,025 ==> BusukBuah	17,889 ==> BusukBuah	18,358 ==> BusukBuah	16 ==> BusukBuah	18,762 ==> BusukBuah
[ 89 ]	17,889 ==> BusukBuah	15,46 ==> BusukBuah	16 ==> BusukBuah	13,229 ==> BusukBuah	16,462 ==> BusukBuah
[ 90 ]	20,396 ==> BusukBuah	18,303 ==> BusukBuah	18,762 ==> BusukBuah	16,462 ==> BusukBuah	19,157 ==> BusukBuah
[ 91 ]	18,303 ==> BusukBuah	15,937 ==> BusukBuah	16,462 ==> BusukBuah	13,784 ==> BusukBuah	16,912 ==> BusukBuah
[ 92 ]	18,762 ==> BusukBuah	16,462 ==> BusukBuah	16,971 ==> BusukBuah	14,387 ==> BusukBuah	17,407 ==> BusukBuah
[ 93 ]	16,462 ==> BusukBuah	13,784 ==> BusukBuah	14,387 ==> BusukBuah	11,225 ==> BusukBuah	14,9 ==> BusukBuah
[ 94 ]	20,976 ==> BusukBuah	18,947 ==> BusukBuah	19,391 ==> BusukBuah	17,176 ==> BusukBuah	19,774 ==> BusukBuah
[ 95 ]	18,947 ==> BusukBuah	16,673 ==> BusukBuah	17,176 ==> BusukBuah	14,829 ==> BusukBuah	17,607 ==> BusukBuah
[ 96 ]	19,391 ==> BusukBuah	17,176 ==> BusukBuah	17,664 ==> BusukBuah	15,199 ==> BusukBuah	18,083 ==> BusukBuah
[ 97 ]	17,176 ==> BusukBuah	14,629 ==> BusukBuah	15,199 ==> BusukBuah	12,247 ==> BusukBuah	15,684 ==> BusukBuah
[ 98 ]	19,774 ==> BusukBuah	17,607 ==> BusukBuah	18,083 ==> BusukBuah	15,684 ==> BusukBuah	18,493 ==> BusukBuah
[ 99 ]	17,607 ==> BusukBuah	15,133 ==> BusukBuah	15,684 ==> BusukBuah	12,845 ==> BusukBuah	16,155 ==> BusukBuah
[ 100 ]	18,083 ==> BusukBuah	15,684 ==> BusukBuah	16,217 ==> BusukBuah	13,491 ==> BusukBuah	16,673 ==> BusukBuah

Gambar 5.3 Implementasi Antarmuka Jarak Euclidian Sebelum Diurutkan

**DIAGNOSA PENYAKIT PADA TANAMAN CABAI**  
MENGUNAKAN METODE FUZZY-KNN (F-KNN)

DATA PROSES FKNN HASIL KLASIFIKASI DAN AKURASI

PARAMETER

NILAI K-

JUMLAH DATA UJI

JUMLAH DATA LATIH

PROSES

D-EUCLIDEAN K-DATA PROSES FUZZY

SEBELUM DIURUT SETELAH DIURUT

[ 78 ]	5 ==> VirusGemini	8,602 ==> VirusGemini	9,434 ==> VirusGemini	10,296 ==> VirusGemini	13,038 ==> VirusGemini
[ 79 ]	5 ==> VirusGemini	9,434 ==> VirusGemini	10,296 ==> VirusGemini	12,042 ==> BercakDaun	12,45 ==> VirusGemini
[ 80 ]	5 ==> VirusGemini	9,434 ==> VirusGemini	10,296 ==> VirusGemini	11,747 ==> VirusGemini	12,728 ==> BercakDaun
[ 81 ]	5 ==> VirusGemini	9 ==> BercakDaun	9,434 ==> VirusGemini	10,296 ==> VirusGemini	11,402 ==> BercakDaun
[ 82 ]	5 ==> VirusGemini	8,602 ==> VirusGemini	11,747 ==> VirusGemini	12,45 ==> VirusGemini	13,304 ==> BercakDaun
[ 83 ]	8,602 ==> VirusGemini	9,798 ==> BercakDaun	10,296 ==> VirusGemini	11,747 ==> VirusGemini	12,042 ==> BercakDaun
[ 84 ]	8,602 ==> VirusGemini	9,434 ==> VirusGemini	10,63 ==> BercakDaun	11,747 ==> VirusGemini	12,45 ==> VirusGemini
[ 85 ]	5,667 ==> BercakDaun	8,602 ==> VirusGemini	9 ==> BercakDaun	10,63 ==> BercakDaun	10,817 ==> BusukBuah
[ 86 ]	4 ==> VirusGemini	8,062 ==> VirusGemini	8,944 ==> VirusGemini	9,849 ==> VirusGemini	12,689 ==> VirusGemini
[ 87 ]	4 ==> VirusGemini	8,944 ==> VirusGemini	9,849 ==> VirusGemini	12,083 ==> VirusGemini	12,41 ==> BercakDaun
[ 88 ]	4 ==> VirusGemini	8,944 ==> VirusGemini	9,849 ==> VirusGemini	11,358 ==> VirusGemini	12,689 ==> VirusGemini
[ 89 ]	4 ==> VirusGemini	8,944 ==> VirusGemini	9,487 ==> BercakDaun	9,849 ==> VirusGemini	11,79 ==> BercakDaun
[ 90 ]	4 ==> VirusGemini	8,062 ==> VirusGemini	11,358 ==> VirusGemini	12,083 ==> VirusGemini	13,638 ==> BercakDaun
[ 91 ]	8,062 ==> VirusGemini	8,949 ==> VirusGemini	10,247 ==> BercakDaun	11,358 ==> VirusGemini	12,083 ==> BercakDaun
[ 92 ]	8,062 ==> VirusGemini	8,944 ==> VirusGemini	11,045 ==> BercakDaun	11,358 ==> VirusGemini	12,083 ==> BercakDaun
[ 93 ]	6,403 ==> BercakDaun	8,062 ==> VirusGemini	9,487 ==> BercakDaun	11,045 ==> BercakDaun	11,225 ==> BercakDaun
[ 94 ]	6,403 ==> VirusGemini	9,487 ==> VirusGemini	10,247 ==> VirusGemini	11,045 ==> VirusGemini	13,638 ==> BercakDaun
[ 95 ]	6,403 ==> VirusGemini	10,247 ==> VirusGemini	11,045 ==> VirusGemini	11,358 ==> BercakDaun	13,077 ==> BercakDaun
[ 96 ]	6,403 ==> VirusGemini	10,247 ==> VirusGemini	11,045 ==> VirusGemini	12,083 ==> BercakDaun	12,41 ==> BercakDaun
[ 97 ]	6,403 ==> VirusGemini	8,062 ==> BercakDaun	10,247 ==> VirusGemini	10,677 ==> BercakDaun	11,045 ==> BercakDaun
[ 98 ]	6,403 ==> VirusGemini	9,487 ==> VirusGemini	12,41 ==> VirusGemini	12,689 ==> BercakDaun	13,077 ==> BercakDaun
[ 99 ]	8,944 ==> BercakDaun	9,487 ==> VirusGemini	11,045 ==> VirusGemini	11,358 ==> BercakDaun	12,41 ==> BercakDaun
[ 100 ]	9,487 ==> VirusGemini	9,849 ==> BercakDaun	10,247 ==> VirusGemini	12,083 ==> BercakDaun	12,41 ==> BercakDaun

Gambar 5.4 Implementasi Antarmuka Jarak Euclidian Setelah Diurutkan

### DIAGNOSA PENYAKIT PADA TANAMAN CABAI

MENGUNAKAN METODE FUZZY-KNN (F-KNN)

DATA
PROSES FKNN
HASIL KLASIFIKASI DAN AKURASI

---

PARAMETER

NILAI K-

JUMLAH DATA UJI

JUMLAH DATA LATIH

PROSES

D-EUCLIDEAN
K-DATA
PROSES FUZZY

[ 76 ] 7 ==> VirusGemini	8 ==> VirusGemini	10,63 ==> VirusGemini	11,402 ==> VirusGemini	11,747 ==> BercakDaun
[ 77 ] 7 ==> VirusGemini	7,55 ==> BercakDaun	10,296 ==> BercakDaun	10,63 ==> VirusGemini	11,402 ==> VirusGem
[ 78 ] 5 ==> VirusGemini	8,602 ==> VirusGemini	9,434 ==> VirusGemini	10,296 ==> VirusGemini	13,038 ==> VirusGe
[ 79 ] 5 ==> VirusGemini	9,434 ==> VirusGemini	10,296 ==> VirusGemini	12,042 ==> BercakDaun	12,45 ==> VirusGe
[ 80 ] 5 ==> VirusGemini	9,434 ==> VirusGemini	10,296 ==> VirusGemini	11,747 ==> VirusGemini	12,728 ==> Bercak
[ 81 ] 5 ==> VirusGemini	9 ==> BercakDaun	9,434 ==> VirusGemini	10,296 ==> VirusGemini	11,402 ==> BercakDaun
[ 82 ] 5 ==> VirusGemini	8,602 ==> VirusGemini	11,747 ==> VirusGemini	12,45 ==> VirusGemini	13,304 ==> BercakD
[ 83 ] 8,602 ==> VirusGemini	9,798 ==> BercakDaun	10,296 ==> VirusGemini	11,747 ==> VirusGemini	12,042 ==> Be
[ 84 ] 8,602 ==> VirusGemini	9,434 ==> VirusGemini	10,63 ==> BercakDaun	11,747 ==> VirusGemini	12,45 ==> Virus
[ 85 ] 5,657 ==> BercakDaun	8,602 ==> VirusGemini	9 ==> BercakDaun	10,63 ==> BercakDaun	10,817 ==> BusukBuz
[ 86 ] 4 ==> VirusGemini	8,062 ==> VirusGemini	8,944 ==> VirusGemini	9,849 ==> VirusGemini	12,689 ==> VirusGem
[ 87 ] 4 ==> VirusGemini	8,944 ==> VirusGemini	9,849 ==> VirusGemini	12,083 ==> VirusGemini	12,41 ==> BercakDa
[ 88 ] 4 ==> VirusGemini	8,944 ==> VirusGemini	9,849 ==> VirusGemini	11,358 ==> VirusGemini	12,689 ==> VirusGe
[ 89 ] 4 ==> VirusGemini	8,944 ==> VirusGemini	9,487 ==> BercakDaun	9,849 ==> VirusGemini	11,79 ==> BercakDau
[ 90 ] 4 ==> VirusGemini	8,062 ==> VirusGemini	11,358 ==> VirusGemini	12,083 ==> VirusGemini	13,638 ==> Bercak
[ 91 ] 8,062 ==> VirusGemini	9,849 ==> VirusGemini	10,247 ==> BercakDaun	11,358 ==> VirusGemini	12,083 ==> Vir
[ 92 ] 8,062 ==> VirusGemini	8,944 ==> VirusGemini	11,045 ==> BercakDaun	11,358 ==> VirusGemini	12,083 ==> Vir
[ 93 ] 6,403 ==> BercakDaun	8,062 ==> VirusGemini	9,487 ==> BercakDaun	11,045 ==> BercakDaun	11,225 ==> Bus
[ 94 ] 6,403 ==> VirusGemini	9,487 ==> VirusGemini	10,247 ==> VirusGemini	11,045 ==> VirusGemini	13,638 ==> Vir
[ 95 ] 6,403 ==> VirusGemini	10,247 ==> VirusGemini	11,045 ==> VirusGemini	11,358 ==> BercakDaun	13,077 ==> V
[ 96 ] 6,403 ==> VirusGemini	10,247 ==> VirusGemini	11,045 ==> VirusGemini	12,083 ==> BercakDaun	12,41 ==> Vir
[ 97 ] 6,403 ==> VirusGemini	8,062 ==> BercakDaun	10,247 ==> VirusGemini	10,677 ==> BercakDaun	11,045 ==> Vir
[ 98 ] 6,403 ==> VirusGemini	9,487 ==> VirusGemini	12,41 ==> VirusGemini	12,689 ==> BercakDaun	13,077 ==> Viru
[ 99 ] 8,944 ==> BercakDaun	9,487 ==> VirusGemini	11,045 ==> VirusGemini	11,358 ==> BercakDaun	12,41 ==> Viru
[ 100 ] 9,487 ==> VirusGemini	9,849 ==> BercakDaun	10,247 ==> VirusGemini	12,083 ==> BercakDaun	12,41 ==> Vir

Gambar 5.5 Implementasi Antarmuka Jarak Euclediane Sebanyak K

### MENGUNAKAN METODE FUZZY-KNN (F-KNN)

DATA
PROSES FKNN
HASIL KLASIFIKASI DAN AKURASI

---

PARAMETER

NILAI K-

JUMLAH DATA UJI

JUMLAH DATA LATIH

PROSES

D-EUCLIDEAN
K-DATA
PROSES FUZZY

FUZZYFIKASI
DEFUZZYFIKASI

No.	Fuzzyfikasi Y	Fuzzyfikasi N
1	0.6325000000000001	0.1225
2	0.6325000000000001	0.1225
3	0.6325000000000001	0.1225
4	0.6325000000000001	0.1225

Gambar 5.6 Implementasi Antarmuka Fuzzifikasi



### DIAGNOSA PENYAKIT PADA TANAMAN CABAI

MENGUNAKAN METODE FUZZY-KNN (F-KNN)

DATA
PROSES FKNN
HASIL KLASIFIKASI DAN AKURASI

---

PARAMETER

NILAI K-

JUMLAH DATA UJI

JUMLAH DATA LATIH

PROSES

D-EUCLIDEAN
K-DATA
PROSES FUZZY

FUZZYFIKASI
DEFUZZYFIKASI

HASIL DEFUZZYFIKASI
HASIL DEFUZZYFIKASI TERURUT

[ 81 ]	0,122 ==> BusukBuah	0,249 ==> BercakDaun	0,122 ==> LayuFusarium	0,506 ==> VirusGemini
[ 82 ]	0,122 ==> BusukBuah	0,162 ==> BercakDaun	0,122 ==> LayuFusarium	0,593 ==> VirusGemini
[ 83 ]	0,122 ==> BusukBuah	0,308 ==> BercakDaun	0,122 ==> LayuFusarium	0,447 ==> VirusGemini
[ 84 ]	0,122 ==> BusukBuah	0,218 ==> BercakDaun	0,122 ==> LayuFusarium	0,537 ==> VirusGemini
[ 85 ]	0,181 ==> BusukBuah	0,481 ==> BercakDaun	0,122 ==> LayuFusarium	0,215 ==> VirusGemini
[ 86 ]	0,123 ==> BusukBuah	0,123 ==> BercakDaun	0,123 ==> LayuFusarium	0,633 ==> VirusGemini
[ 87 ]	0,122 ==> BusukBuah	0,156 ==> BercakDaun	0,122 ==> LayuFusarium	0,599 ==> VirusGemini
[ 88 ]	0,122 ==> BusukBuah	0,122 ==> BercakDaun	0,122 ==> LayuFusarium	0,633 ==> VirusGemini
[ 89 ]	0,122 ==> BusukBuah	0,213 ==> BercakDaun	0,122 ==> LayuFusarium	0,542 ==> VirusGemini
[ 90 ]	0,123 ==> BusukBuah	0,151 ==> BercakDaun	0,123 ==> LayuFusarium	0,504 ==> VirusGemini
[ 91 ]	0,122 ==> BusukBuah	0,22 ==> BercakDaun	0,122 ==> LayuFusarium	0,535 ==> VirusGemini
[ 92 ]	0,122 ==> BusukBuah	0,205 ==> BercakDaun	0,122 ==> LayuFusarium	0,55 ==> VirusGemini
[ 93 ]	0,183 ==> BusukBuah	0,455 ==> BercakDaun	0,122 ==> LayuFusarium	0,24 ==> VirusGemini
[ 94 ]	0,122 ==> BusukBuah	0,122 ==> BercakDaun	0,122 ==> LayuFusarium	0,632 ==> VirusGemini
[ 95 ]	0,122 ==> BusukBuah	0,193 ==> BercakDaun	0,122 ==> LayuFusarium	0,562 ==> VirusGemini
[ 96 ]	0,122 ==> BusukBuah	0,185 ==> BercakDaun	0,122 ==> LayuFusarium	0,57 ==> VirusGemini
[ 97 ]	0,122 ==> BusukBuah	0,308 ==> BercakDaun	0,122 ==> LayuFusarium	0,447 ==> VirusGemini
[ 98 ]	0,122 ==> BusukBuah	0,181 ==> BercakDaun	0,122 ==> LayuFusarium	0,574 ==> VirusGemini
[ 99 ]	0,123 ==> BusukBuah	0,347 ==> BercakDaun	0,123 ==> LayuFusarium	0,408 ==> VirusGemini
[ 100 ]	0,122 ==> BusukBuah	0,32 ==> BercakDaun	0,122 ==> LayuFusarium	0,435 ==> VirusGemini

**Gambar 5.7 Implementasi Antarmuka Defuzzifikasi**

#### 5.4.3 Implementasi Hasil Klasifikasi dan Akurasi

Pada implementasi hasil klasifikasi dan data uji ini akan ditampilkan nilai akhir proses FKNN, hasil klasifikasi dan tingkat akurasi sistem berdasarkan data uji yang dimasukkan pengguna. Implementasi antarmuka hasil klasifikasi dan akurasi ditunjukkan oleh gambar 5.8.

### DIAGNOSA PENYAKIT PADA TANAMAN CABAI

MENGUNAKAN METODE FUZZY-KNN (F-KNN)

DATA
PROSES FKNN
HASIL KLASIFIKASI DAN AKURASI

---

No.	Nilai Fuzzy	Hasil Klasifikasi	Kelas Asli
1	0.6325000000000001	BusukBuah	BusukBuah
2	0.6325000000000001	BusukBuah	BusukBuah
3	0.6325000000000001	BusukBuah	BusukBuah
4	0.6320957448313091	BusukBuah	BusukBuah
5	0.6325000000000001	BusukBuah	BusukBuah
6	0.5645637096472282	BusukBuah	BusukBuah
7	0.5747664117698896	BusukBuah	BusukBuah
8	0.4355166230946004	BusukBuah	BusukBuah
9	0.6325	BusukBuah	BusukBuah
10	0.5731327016944927	BusukBuah	BusukBuah
11	0.5793299367472864	BusukBuah	BusukBuah
12	0.47850744132799095	BusukBuah	BusukBuah
13	0.5827594512431953	BusukBuah	BusukBuah
14	0.4333464176799826	BusukBuah	BusukBuah
15	0.45636932154309906	BusukBuah	BusukBuah
16	0.48569462891665244	BercakDaun	BusukBuah
17	0.6325000000000003	BusukBuah	BusukBuah
18	0.6325000000000001	BusukBuah	BusukBuah
19	0.6325000000000002	BusukBuah	BusukBuah
20	0.5495616495199992	BusukBuah	BusukBuah
21	0.6059647484999995	BusukBuah	BusukBuah
22	0.5426200747280526	BusukBuah	BusukBuah
23	0.5558154260070645	BusukBuah	BusukBuah
24	0.3692775354429307	BercakDaun	BusukBuah
25	0.6324999999999998	BusukBuah	BusukBuah
26	0.4926019094348792	BusukBuah	BusukBuah
27	0.5601483455231063	BusukBuah	BusukBuah
28	0.42553712224430995	BusukBuah	BusukBuah
29	0.5649555755475123	BusukBuah	BusukBuah
30	0.3563601656789959	BercakDaun	BusukBuah
31	0.42285222054008205	BusukBuah	BusukBuah

#### AKURASI

92%

Keterangan :

Jumlah Data Latih : **20**

Jumlah Data Uji : **100**

Jumlah Diagnosa Benar : **92**

Jumlah Diagnosa Salah : **8**

Perhitungan Akurasi  
Diagnosa Benar/ Jlh Data Uji \* 100%

**Gambar 5.8 Implementasi Antarmuka Klasifikasi dan Akurasi**

## BAB 6 PENGUJIAN DAN ANALISIS

Bagian ini menjelaskan tentang pengujian dan analisis dari implementasi F-KNN untuk diagnosis penyakit pada tanaman cabai. Adapun pengujian yang dilakukan adalah pengujian akurasi sistem.

### 6.1 Pengujian Akurasi Sistem

Pengujian akurasi sistem dilakukan dengan memasukan nilai k yang bervariasi dengan nilai 5,10,15 dan 20. Jumlah data uji yang digunakan sebanyak 100 data, Data uji ditunjukkan oleh table 6.1.

**Tabel 6.1 Data Uji 1**

PENYAKIT	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G46
BusukBuah	4	6	7	8	9	0	0	0	0	-	-	-	0
BusukBuah	4	6	7	8	0	0	0	0	0	-	-	-	0
BusukBuah	4	6	7	0	9	0	0	0	0	-	-	-	0
BusukBuah	4	6	7	0	0	0	0	0	0	-	-	-	0
BusukBuah	4	6	0	8	9	0	0	0	0	-	-	-	0
BusukBuah	4	6	0	8	0	0	0	0	0	-	-	-	0
BusukBuah	4	6	0	0	9	0	0	0	0	-	-	-	0
BusukBuah	4	6	0	0	0	0	0	0	0	-	-	-	0
BusukBuah	4	0	7	8	9	0	0	0	0	-	-	-	0
BusukBuah	4	0	7	8	0	0	0	0	0	-	-	-	0
BusukBuah	4	0	7	0	9	0	0	0	0	-	-	-	0
BusukBuah	4	0	7	0	0	0	0	0	0	-	-	-	0
BusukBuah	4	0	0	8	9	0	0	0	0	-	-	-	0
BusukBuah	4	0	0	8	0	0	0	0	0	-	-	-	0
BusukBuah	4	0	0	0	9	0	0	0	0	-	-	-	0
BusukBuah	4	0	0	0	0	0	0	0	0	-	-	-	0
BusukBuah	0	6	7	8	9	0	0	0	0	-	-	-	0
BusukBuah	0	6	7	8	0	0	0	0	0	-	-	-	0
BusukBuah	0	6	7	0	9	0	0	0	0	-	-	-	0
BusukBuah	0	6	7	0	0	0	0	0	0	-	-	-	0
BusukBuah	0	6	0	8	9	0	0	0	0	-	-	-	0
BusukBuah	0	6	0	8	0	0	0	0	0	-	-	-	0

BusukBuah	0	6	0	0	9	0	0	0	0	-	-	-	0
BusukBuah	0	6	0	0	0	0	0	0	0	-	-	-	0
BusukBuah	0	0	7	8	9	0	0	0	0	-	-	-	0
BusukBuah	0	0	7	8	0	0	0	0	0	-	-	-	0
BusukBuah	0	0	7	0	9	0	0	0	0	-	-	-	0
BusukBuah	0	0	7	0	0	0	0	0	0	-	-	-	0
BusukBuah	0	0	0	8	9	0	0	0	0	-	-	-	0
BusukBuah	0	0	0	8	0	0	0	0	0	-	-	-	0
BusukBuah	0	0	0	0	9	0	0	0	0	-	-	-	0
BercakDaun	0	0	0	0	0	4	7	9	0	-	-	-	0
BercakDaun	0	0	0	0	0	4	7	0	0	-	-	-	0
BercakDaun	0	0	0	0	0	4	0	9	0	-	-	-	0
BercakDaun	0	0	0	0	0	4	0	0	0	-	-	-	0
BercakDaun	0	0	0	0	0	0	7	9	0	-	-	-	0
BercakDaun	0	0	0	0	0	0	7	0	0	-	-	-	0
BercakDaun	0	0	0	0	0	0	0	9	0	-	-	-	0
LayuFusarium	0	0	0	0	0	0	0	0	5	-	-	-	4
LayuFusarium	0	0	0	0	0	0	0	0	5	-	-	-	0
LayuFusarium	0	0	0	0	0	0	0	0	5	-	-	-	4
LayuFusarium	0	0	0	0	0	0	0	0	5	-	-	-	0
LayuFusarium	0	0	0	0	0	0	0	0	5	-	-	-	4
LayuFusarium	0	0	0	0	0	0	0	0	5	-	-	-	0
LayuFusarium	0	0	0	0	0	0	0	0	5	-	-	-	4
LayuFusarium	0	0	0	0	0	0	0	0	5	-	-	-	0
LayuFusarium	0	0	0	0	0	0	0	0	5	-	-	-	4
LayuFusarium	0	0	0	0	0	0	0	0	5	-	-	-	0
LayuFusarium	0	0	0	0	0	0	0	0	5	-	-	-	4
LayuFusarium	0	0	0	0	0	0	0	0	5	-	-	-	0
LayuFusarium	0	0	0	0	0	0	0	0	5	-	-	-	4
LayuFusarium	0	0	0	0	0	0	0	0	5	-	-	-	0
LayuFusarium	0	0	0	0	0	0	0	0	5	-	-	-	4
LayuFusarium	0	0	0	0	0	0	0	0	5	-	-	-	0
LayuFusarium	0	0	0	0	0	0	0	0	5	-	-	-	4
LayuFusarium	0	0	0	0	0	0	0	0	5	-	-	-	0



[illegible]

VirusGemini	0	0	0	0	0	0	0	0	0	-	-	-	0
VirusGemini	0	0	0	0	0	0	0	0	0	-	-	-	0
VirusGemini	0	0	0	0	0	0	0	0	0	-	-	-	0
VirusGemini	0	0	0	0	0	0	0	0	0	-	-	-	0
VirusGemini	0	0	0	0	0	0	0	0	0	-	-	-	0
VirusGemini	0	0	0	0	0	0	0	0	0	-	-	-	0
VirusGemini	0	0	0	0	0	0	0	0	0	-	-	-	0
VirusGemini	0	0	0	0	0	0	0	0	0	-	-	-	0
VirusGemini	0	0	0	0	0	0	0	0	0	-	-	-	0
VirusGemini	0	0	0	0	0	0	0	0	0	-	-	-	0
VirusGemini	0	0	0	0	0	0	0	0	0	-	-	-	0
VirusGemini	0	0	0	0	0	0	0	0	0	-	-	-	0
VirusGemini	0	0	0	0	0	0	0	0	0	-	-	-	0
VirusGemini	0	0	0	0	0	0	0	0	0	-	-	-	0
VirusGemini	0	0	0	0	0	0	0	0	0	-	-	-	0
VirusGemini	0	0	0	0	0	0	0	0	0	-	-	-	0

### 6.1.1 Pengujian dengan nilai K = 5

Pengujian yang pertama adalah pengujian dengan nilai K = 5, data uji yang digunakan sebanyak 100 data uji. Hasil pengujian dengan nilai K = 5 ditunjukkan oleh tabel 6.2.

**Tabel 6.2 Hasil Uji K = 5**

No	Hasil Sistem	Kelas Asli
1	BusukBuah	BusukBuah
2	BusukBuah	BusukBuah
3	BusukBuah	BusukBuah
4	BusukBuah	BusukBuah
5	BusukBuah	BusukBuah
6	BusukBuah	BusukBuah
7	BusukBuah	BusukBuah
8	BusukBuah	BusukBuah
9	BusukBuah	BusukBuah
10	BusukBuah	BusukBuah
11	BusukBuah	BusukBuah
12	BusukBuah	BusukBuah
13	BusukBuah	BusukBuah
14	BusukBuah	BusukBuah
15	BusukBuah	BusukBuah
16	BercakDaun	BusukBuah

17	BusukBuah	BusukBuah
18	BusukBuah	BusukBuah
19	BusukBuah	BusukBuah
20	BusukBuah	BusukBuah
21	BusukBuah	BusukBuah
22	BusukBuah	BusukBuah
23	BusukBuah	BusukBuah
24	BercakDaun	BusukBuah
25	BusukBuah	BusukBuah
26	BusukBuah	BusukBuah
27	BusukBuah	BusukBuah
28	BusukBuah	BusukBuah
29	BusukBuah	BusukBuah
30	BercakDaun	BusukBuah
31	BusukBuah	BusukBuah
32	BercakDaun	BercakDaun
33	BercakDaun	BercakDaun
34	BercakDaun	BercakDaun
35	BercakDaun	BercakDaun
36	BercakDaun	BercakDaun
37	BercakDaun	BercakDaun
38	BercakDaun	BercakDaun
39	LayuFusarium	LayuFusarium
40	LayuFusarium	LayuFusarium
41	LayuFusarium	LayuFusarium
42	LayuFusarium	LayuFusarium
43	LayuFusarium	LayuFusarium
44	LayuFusarium	LayuFusarium
45	LayuFusarium	LayuFusarium
46	LayuFusarium	LayuFusarium
47	LayuFusarium	LayuFusarium
48	LayuFusarium	LayuFusarium
49	LayuFusarium	LayuFusarium
50	LayuFusarium	LayuFusarium
51	LayuFusarium	LayuFusarium
52	LayuFusarium	LayuFusarium
53	LayuFusarium	LayuFusarium
54	BercakDaun	LayuFusarium
55	LayuFusarium	LayuFusarium
56	LayuFusarium	LayuFusarium
57	LayuFusarium	LayuFusarium
58	LayuFusarium	LayuFusarium

59	LayuFusarium	LayuFusarium
60	LayuFusarium	LayuFusarium
61	LayuFusarium	LayuFusarium
62	BercakDaun	LayuFusarium
63	LayuFusarium	LayuFusarium
64	LayuFusarium	LayuFusarium
65	LayuFusarium	LayuFusarium
66	LayuFusarium	LayuFusarium
67	LayuFusarium	LayuFusarium
68	LayuFusarium	LayuFusarium
69	BercakDaun	LayuFusarium
70	VirusGemini	VirusGemini
71	VirusGemini	VirusGemini
72	VirusGemini	VirusGemini
73	VirusGemini	VirusGemini
74	VirusGemini	VirusGemini
75	VirusGemini	VirusGemini
76	VirusGemini	VirusGemini
77	VirusGemini	VirusGemini
78	VirusGemini	VirusGemini
79	VirusGemini	VirusGemini
80	VirusGemini	VirusGemini
81	VirusGemini	VirusGemini
82	VirusGemini	VirusGemini
83	VirusGemini	VirusGemini
84	VirusGemini	VirusGemini
85	BercakDaun	VirusGemini
86	VirusGemini	VirusGemini
87	VirusGemini	VirusGemini
88	VirusGemini	VirusGemini
89	VirusGemini	VirusGemini
90	VirusGemini	VirusGemini
91	VirusGemini	VirusGemini
92	VirusGemini	VirusGemini
93	BercakDaun	VirusGemini
94	VirusGemini	VirusGemini
95	VirusGemini	VirusGemini
96	VirusGemini	VirusGemini
97	VirusGemini	VirusGemini
98	VirusGemini	VirusGemini
99	VirusGemini	VirusGemini
100	VirusGemini	VirusGemini

Dari hasil pengujian diatas didapatkan sebanyak 8 data uji yang memiliki hasil klasifikasi berbeda dengan kelas sebenarnya, dari hasil tersebut kemudian akan dihitung akurasi menggunakan persamaan (2-8) sehingga dihasilkan akurasi sebagai berikut :

$$\begin{aligned} Akurasi &= \frac{92}{100} \times 100\% \\ &= 92\% \end{aligned}$$

### 6.1.2 Pengujian dengan nilai K = 10

Pengujian yang pertama adalah pengujian dengan nilai K = 10, data uji yang digunakan sebanyak 100 data uji. Hasil pengujian dengan nilai K = 10 ditunjukkan oleh tabel 6.3.

**Tabel 6.3 Hasil Uji K = 10**

No	Hasil Sistem	Kelas Asli
1	BusukBuah	BusukBuah
2	BusukBuah	BusukBuah
3	BusukBuah	BusukBuah
4	BusukBuah	BusukBuah
5	BusukBuah	BusukBuah
6	BusukBuah	BusukBuah
7	BusukBuah	BusukBuah
8	BusukBuah	BusukBuah
9	BusukBuah	BusukBuah
10	BusukBuah	BusukBuah
11	BusukBuah	BusukBuah
12	BusukBuah	BusukBuah
13	BusukBuah	BusukBuah
14	BusukBuah	BusukBuah
15	BusukBuah	BusukBuah
16	BercakDaun	BusukBuah
17	BusukBuah	BusukBuah
18	BusukBuah	BusukBuah
19	BusukBuah	BusukBuah
20	BusukBuah	BusukBuah
21	BusukBuah	BusukBuah
22	BusukBuah	BusukBuah
23	BusukBuah	BusukBuah
24	BercakDaun	BusukBuah
25	BusukBuah	BusukBuah
26	BusukBuah	BusukBuah
27	BusukBuah	BusukBuah
28	BercakDaun	BusukBuah

29	BusukBuah	BusukBuah
30	BercakDaun	BusukBuah
31	BusukBuah	BusukBuah
32	BercakDaun	BercakDaun
33	BercakDaun	BercakDaun
34	BercakDaun	BercakDaun
35	BercakDaun	BercakDaun
36	BercakDaun	BercakDaun
37	BercakDaun	BercakDaun
38	BercakDaun	BercakDaun
39	LayuFusarium	LayuFusarium
40	LayuFusarium	LayuFusarium
41	LayuFusarium	LayuFusarium
42	LayuFusarium	LayuFusarium
43	LayuFusarium	LayuFusarium
44	LayuFusarium	LayuFusarium
45	LayuFusarium	LayuFusarium
46	LayuFusarium	LayuFusarium
47	LayuFusarium	LayuFusarium
48	LayuFusarium	LayuFusarium
49	LayuFusarium	LayuFusarium
50	LayuFusarium	LayuFusarium
51	LayuFusarium	LayuFusarium
52	LayuFusarium	LayuFusarium
53	BercakDaun	LayuFusarium
54	BercakDaun	LayuFusarium
55	LayuFusarium	LayuFusarium
56	LayuFusarium	LayuFusarium
57	LayuFusarium	LayuFusarium
58	LayuFusarium	LayuFusarium
59	LayuFusarium	LayuFusarium
60	LayuFusarium	LayuFusarium
61	LayuFusarium	LayuFusarium
62	BercakDaun	LayuFusarium
63	LayuFusarium	LayuFusarium
64	LayuFusarium	LayuFusarium
65	LayuFusarium	LayuFusarium
66	LayuFusarium	LayuFusarium
67	LayuFusarium	LayuFusarium
68	LayuFusarium	LayuFusarium
69	BercakDaun	LayuFusarium
70	VirusGemini	VirusGemini

71	VirusGemini	VirusGemini
72	VirusGemini	VirusGemini
73	VirusGemini	VirusGemini
74	VirusGemini	VirusGemini
75	VirusGemini	VirusGemini
76	VirusGemini	VirusGemini
77	VirusGemini	VirusGemini
78	VirusGemini	VirusGemini
79	VirusGemini	VirusGemini
80	VirusGemini	VirusGemini
81	VirusGemini	VirusGemini
82	VirusGemini	VirusGemini
83	VirusGemini	VirusGemini
84	VirusGemini	VirusGemini
85	BercakDaun	VirusGemini
86	VirusGemini	VirusGemini
87	VirusGemini	VirusGemini
88	VirusGemini	VirusGemini
89	VirusGemini	VirusGemini
90	VirusGemini	VirusGemini
91	VirusGemini	VirusGemini
92	VirusGemini	VirusGemini
93	BercakDaun	VirusGemini
94	VirusGemini	VirusGemini
95	VirusGemini	VirusGemini
96	VirusGemini	VirusGemini
97	VirusGemini	VirusGemini
98	VirusGemini	VirusGemini
99	VirusGemini	VirusGemini
100	VirusGemini	VirusGemini

Dari hasil pengujian diatas didapatkan sebanyak 10 data uji yang memiliki hasil klasifikasi berbeda dengan kelas sebenarnya, dari hasil tersebut kemudian akan dihitung akurasi menggunakan persamaan (2-8) sehingga dihasilkan akurasi sebagai berikut :

$$\begin{aligned}
 Akurasi &= \frac{90}{100} \times 100\% \\
 &= 90\%
 \end{aligned}$$

### 6.1.3 Pengujian dengan nilai K = 15

Pengujian yang pertama adalah pengujian dengan nilai K = 15, data uji yang digunakan sebanyak 100 data uji. Hasil pengujian dengan nilai K = 15 ditunjukkan oleh tabel 6.4.

Tabel 6.4 Data Uji K = 15

No	Hasil Sistem	Kelas Asli
1	BusukBuah	BusukBuah
2	BusukBuah	BusukBuah
3	BusukBuah	BusukBuah
4	BusukBuah	BusukBuah
5	BusukBuah	BusukBuah
6	BusukBuah	BusukBuah
7	BusukBuah	BusukBuah
8	BusukBuah	BusukBuah
9	BusukBuah	BusukBuah
10	BusukBuah	BusukBuah
11	BusukBuah	BusukBuah
12	BusukBuah	BusukBuah
13	BusukBuah	BusukBuah
14	BusukBuah	BusukBuah
15	BusukBuah	BusukBuah
16	BercakDaun	BusukBuah
17	BusukBuah	BusukBuah
18	BusukBuah	BusukBuah
19	BusukBuah	BusukBuah
20	BusukBuah	BusukBuah
21	BusukBuah	BusukBuah
22	BusukBuah	BusukBuah
23	BusukBuah	BusukBuah
24	BercakDaun	BusukBuah
25	BusukBuah	BusukBuah
26	BusukBuah	BusukBuah
27	BusukBuah	BusukBuah
28	BercakDaun	BusukBuah
29	BusukBuah	BusukBuah
30	BercakDaun	BusukBuah
31	BusukBuah	BusukBuah
32	BercakDaun	BercakDaun
33	BercakDaun	BercakDaun
34	BercakDaun	BercakDaun
35	BercakDaun	BercakDaun
36	BercakDaun	BercakDaun
37	BercakDaun	BercakDaun
38	BercakDaun	BercakDaun
39	LayuFusarium	LayuFusarium
40	LayuFusarium	LayuFusarium















## DAFTAR PUSTAKA

- Agustina, E. 2017. *Identifikasi Penyakit Tanaman Jarak Pagar Menggunakan Metode Fuzzy K-Nearest Neighbor (FK-NN)*. Fakultas Ilmu Komputer, Universitas Brawijaya.
- Angga, A. 2017, *Klasifikasi Kualitas Tanaman Cabai Menggunakan Metode Fuzzy K-Nearest Neighbor (FKNN)*. Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang.
- Basuki, R. S. 1988. *Analisis Biaya dan Pendapatan Usahatani Cabai Merah (Capsicum annuum L.)* di Desa Kemurang Kulon, Brebes. Bul. Penel. Hort. XV(2):294-299.
- Darmawan, D.A. and E. Pasandaran. 2000. Indonesia. In: M. Ali (ed). *Dynamic of vegetable production. distribution and consumption in Asia*. AVRDC Publication 00-498. Shanhua. Tainan: AVRDC. Pp.139-171. <http://www.avrdc.org/pdf/dynamics/Indonesia.pfd>
- Haryanto, T.2006, *Sistem pakar diagnosa penyakit pada ayam*. Samarinda: Jurnal Informasi Mulawarman
- Kusnawi. 2007. *Pengantar Solusi Data Mining*. Seminar Nasional Teknologi 2007 (SNT). Yogyakarta: STMIK AMIKOM Yogyakarta.
- Satria Dwi Nugraha, R. R. 2016. *Penerapan Fuzzy K -Nearest Neighbor (FK -NN) Dalam Menentukan Status Gizi Balita*. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer Vol. 1, No. 9, Juni 2017,, 925 -932.
- Setiadi. 2004. *Bertanam Cabai*. Penebar Swadaya. Jakarta. 12 hlm. Warisno dan Dahana. 2010. *Peluang Usaha dan Budidaya Cabai*. Jakarta: Gramedia Pustaka Utama.
- Setiadi. 2011. *Bertanam Cabai di Lahan dan Pot*. Penebar Swadaya. Jakarta. 21 hlm
- Shouman, M. 2012, *Applying K-Nearest Neighbour in Diagnosing Heart Disease Patients*, International Journal of Information and Education Technology, No, 3, Vol 2.